

(2)

【特許請求の範囲】

【請求項1】 複数のデータをあらかじめ定めた様式で表わしている帳票を格納するデータベース装置において、前記帳票に記載されるデータ項目毎の記載開始位置および記載終了位置と該データ項目のデータ属性およびデータ長を所定の記号で記述した帳票のサンプルを走査してこの所定の記号を判読し、判読した結果をデータベースのスキーマ定義言語に変換して、データベースのスキーマ定義情報を作成することを特徴とするスキーマ定義情報作成方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、ワードプロセッサ等で作成される複数のデータ項目からなる固定フォームの書類（以下「帳票」と略す）のサンプルに記述されている情報に基づいてデータベースのスキーマ定義情報を作成する方法に関する。

【0002】

【従来の技術】 多数の帳票の内容をデータベース化するには、帳票そのものをイメージリーダーあるいはイメージスキャナで読み取りイメージデータとして光ファイルに格納する方法がある。この方法は、データの蓄積は容易であるが、蓄積されたデータはコードデータでなく、計算機による各種のデータ処理に不向きである。このため、蓄積されたデータを直接計算機で検索や統計処理することができるようにするため、帳票の内容をデータ項目毎にコードデータでデータベース化することが考えられる。

【0003】 この場合、データベースを使用するユーザはデータを格納するデータベースの論理的な構造（スキーマ）を定義することが必要である。すなわち、使用するデータベースのスキーマ定義言語を使用してデータベースの格納単位毎にデータ項目名、データ属性およびデータ長を定義するのである。

【0004】 以下、リレーショナルデータベースのスキーマ定義言語（原 潔著、「標準SQLプログラミング」、啓学出版；参考）を使用したテーブル（格納単位）の論理的な構造の定義情報例を以下に示す。

【0005】

CREATE TABLE クレーム情報

```
(
  表題          VARCHAR(32),
  クレーム内容  VARCHAR(432),
  処置状況      VARCHAR(504),
  受付番号      CHAR(8),
  受付組織名    VARCHAR(9),
  受付年月日    DEC(6),
  処置組織名    VARCHAR(9),
  処置年月日    DEC(6)
)
```

この例は「クレーム情報」というテーブルの論理的な構造の定義情報例である。このテーブルは、データ項目が「表題」、「クレーム内容」、「処置状況」、「受付組織名」、「受付番号」、「受付年月日」、「処置組織名」、「処置年月日」からなり、それぞれのデータ属性およびデータ長は「可変長文字列、最大32文字」、「可変長文字列、最大432文字」、「可変長文字列、最大504文字」、「可変長文字列、最大9文字」、「固定長文字列、8文字」、「十進数、6桁」、「可変長文字列、最大9文字」、「十進数、6桁」であることを定義している。スキーマ定義情報からデータベースの実際の格納領域を用意する操作はデータベース管理システムが実行する。この様にテーブルのスキーマ定義は、帳票フォームのデザインとは別途行う必要があり、使用するデータベース専用のスキーマ定義言語を使用する必要がある。

【0006】

【発明が解決しようとする課題】 以上のように、帳票に記述された内容をコードデータでデータベース化するには、帳票フォームの定義とは別に使用するデータベースのスキーマ定義言語を使用して、データベースの論理的な構造を定義する必要があり、同じような定義を二重にしなければならず、手間を要するという問題があった。

【0007】 本発明は上記に鑑みてなされたもので、その目的としては、帳票に記述された内容のコードデータによるデータベース化を簡易に行なえるようにしたスキーマ定義情報作成方法を提供することにある。

【0008】

【課題を解決するための手段】 上記目的を達成するため、複数のデータをあらかじめ定めた様式で表わしている帳票を格納するデータベース装置において、本発明は、前記帳票に記載されるデータ項目毎の記載開始位置および記載終了位置と該データ項目のデータ属性およびデータ長を所定の記号で記述した帳票のサンプルを走査してこの所定の記号を判読し、判読した結果をデータベースのスキーマ定義言語に変換して、データベースのスキーマ定義情報を作成することを要旨とする。

【0009】

【作用】 本発明のスキーマ定義情報作成方法は、ワード

(3)

プロセッサ等で作成された帳票のサンプルを入力し、サンプルに記述されている所定の記号をデータベースの各データ項目名、データ属性およびデータ長を定義するスキーマ定義情報に変換することで、従来、帳票のフォーム作成とは別に行っていたデータベースのスキーマ定義情報の作成を省略している。

【0010】

【実施例】以下、図面を用いて本発明の実施例を説明する。

【0011】図1は本発明の一実施例のシステム構成例を示すものである。同図において、1はデータベース装置であり、2は帳票のサンプルおよび帳票を作成するワードプロセッサであり、3はデータベース装置1とワードプロセッサ2を接続する通信路である。なお、ここでワードプロセッサ2は専用のワードプロセッサに限らず、パーソナルコンピュータ等のソフトウェアで構成されるワードプロセッサであってもよく、また、通信路3は交換回線やLANであってもよい。

【0012】図2は本実施例に使用する帳票サンプルの1例を示す。図2において、100は帳票サンプルのフォーム（以下、単に「帳票サンプル」と呼ぶ）を示す。101は本帳票例に固定的に記述されている文字であり、データベースのスキーマ定義においてデータ項目名となる。以下、図2において、便宜的にアンダーラインを付与している文字が固定的に記述されている文字であることを示している。102は直前のデータ項目名に対応するテキストデータが記述される開始位置を示す記号である。103は前記テキストデータが記述される終了位置を示す記号である。104は前記テキストデータを固定長データとしてテーブルへ格納するデータ項目のデータ属性とデータ長を示す。記号「C」はデータ属性が固定長文字列であることを示し、「C」の数に2を加えた数がデータ長を示す。記号「D」はデータ属性が十進数であることを示し、「D」の数に2を加えた数がデ

本例における最初の文字は、罫線「┌」であり、さらに罫線「-」・「┐」
「└」・「┘」が続く、次に、「表」、「題」、「」と続く。

「『」を検出したところで、直前の（スペース、罫線以外の）文字「表」、「題」を「列名」に保存する（ステップ230～250）。更に、「『」の次の文字を入力すると「スペース」であり、次に「』」が検出されるまで「スペース」の数をカウントする。本例では30文字であり、「カウント数+2」は32となる。これに基づき次のスキーマ定義情報の1行を作成する（ステップ260～310）。

【0017】

表題 VCHAR(32),

次に、「『」を検出するまで走査を続ける（ステップ400, 410）。そして同様に、直前の「クレーム内

タ長を示す。また、記号「『」102と記号「』」103の間がスペースの場合は、データ属性が可変長文字列であることを示し、スペースの数に2を加えた数がデータ長を示す。ここで、データ長に2を加えるのは、記号「『」102と記号「』」103の位置にもテキストデータの文字が入ることを想定しているからである。105は帳票に固定的に記述されている罫線である。

【0013】次に、本実施例の作用を図2の帳票サンプル100を例に、図3に示す処理フローチャートを用いて説明する。データベースのスキーマ定義情報を作成するに際しては、まず、ワードプロセッサ2により図2に示す帳票のサンプルを作成する必要がある。すなわち、まず、帳票フォームそのものの定義として、図2における符号102, 103, 104で示した記号を除いたものを設計する（以後、データベースの元データとなる帳票は前記フォームを用いて記述する）。次に、データベースのスキーマ定義情報を作成するための入力データとして、図2における符号102, 103, 104で示す記号を追加した帳票サンプル100を作成する。

【0014】次に、ワードプロセッサ2から、スキーマ定義情報作成開始指令とその開始パラメータである表名「クレーム情報」をデータベース装置1へ転送することによりスキーマ定義情報作成動作が開始される。

【0015】開始パラメータとして転送された表名「クレーム情報」は「表名」に保存されており、これに基づき次のスキーマ定義情報の2行を作成する（ステップ200, 210）。

【0016】

CREATE TABLE クレーム情報

(

次に、帳票サンプル100を受け取り、その走査を開始する（ステップ220）

【外1】

容」を「列名」に保存する（ステップ250）。更に、「『」の次の文字を入力すると、「スペース」であり、次に「』」が検出されるまで、罫線をスキップして「スペース」の数をカウントする。本例では430文字であり、「カウント数+2」は432となる。これに基づき次のスキーマ定義情報の1行を作成する（ステップ260～310）。

【0018】

クレーム内容 VCHAR(432),

次に、上記と同様な操作により、次のスキーマ定義情報の1行を作成する（ステップ400, 410, 250～310）。

(4)

【0019】

処置状況 VCHAR(504),

次に、再び「『」を検出するまで走査を続け、同様に、直前の「受付番号」を“列名”に保存する（ステップ400, 410, 250）。ここで、「『」の次の文字を入手すると「C」であり、次に「』」が検出されるまで、「C」の数をカウントする。本例では6文字であり、“カウント+2”は8となる。これに基づき次のスキーマ定義情報の1行を作成する（ステップ260, 270, 320～350）。

【0020】

受付番号 CHAR(8),

以下、同様な操作により、以下のスキーマ定義情報を作成する（ステップ400, 410, 250～310, 360～390）。

【0021】

受付組織名 VCHAR(9),

受付年月日 DEC(6),

処置組織名 VCHAR(9),

処置年月日 DEC(6),

更に、走査を続け、ファイル修正を検出したときに、一時格納されているスキーマ定義情報の最後の行を修正し、以下の2行のスキーマ定義情報を作成する（ステップ400, 410, 250～270, 360～390）。

【0022】

処置年月日 DEC(6),

)

以上の操作により作成され、一時格納されているスキーマ定義情報を以下のように並べて、スキーマ定義情報ファイルに格納する（ステップ420～440）。

【0023】

CREATE TABLE クレーム情報

(

表題 VCHAR(32),

クレーム内容 VCHAR(432),

処置内容 VCHAR(504),

受付番号 CHAR(8),

受付組織名 VCHAR(9),

受付年月日 DEC(6),

処置組織名 VCHAR(9),

処置年月日 DEC(6),

)

以上のスキーマ定義情報をデータベース管理システムに与えると、データベース管理システムはスキーマ定義情報に従って、格納データベースの論理的な構造を定義し、更に物理的な格納領域を用意することになるのである。

【0024】したがって、本実施例によれば、ワードプロセッサ等で作成した帳票サンプルにより、以後入力する帳票に記述されているテキストデータを格納するデータベースのスキーマ定義情報を作成するため、データベースに非熟練の一般的な操作者にもデータベースのスキーマ定義が容易に実行できる。また、帳票フォームのデザインとデータベースのスキーマ設計を兼ねることができ、設計の省力化となる。

【0025】なお、本発明は、上述した実施例で用いた帳票フォームおよび記述記号に限られるものではないことは言うまでもない。

【0026】

【発明の効果】以上説明したように、本発明によれば、ワードプロセッサ等で作成された帳票のサンプルを入力し、サンプルに記述されている所定の記号をデータベースの各データ項目名、データ属性およびデータ長を定義するスキーマ定義情報に変換することで、従来、帳票のフォーム作成とは別に行っていたデータベースのスキーマ定義情報の作成を省略するようにしたので、帳票に記述された内容のコードデータによるデータベース化を簡易に行なうことができ、もって一般的な操作者が使用可能なワードプロセッサを端末としたテキストデータのデータベース格納システムを容易に構築可能とすることができる。

【図面の簡単な説明】

【図1】本発明の一実施例のシステム構成図である。

【図2】当該一実施例を説明するための帳票サンプルを示す図である。

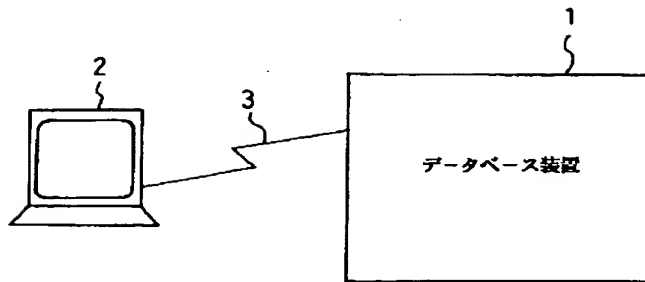
【図3】当該一実施例の処理フローチャートを示す図である。

【符号の説明】

- 1 データベース装置
- 2 ワードプロセッサ
- 3 通信路
- 100 帳票サンプル

(5)

【図1】



【図2】

Figure 2 is a detailed view of a form (100) with various fields and labels. The form is divided into sections for "表題" (Title), "処理内容" (Processing Content), "処理状況" (Processing Status), and a bottom section for "受付番号" (Receipt Number), "受付組織名" (Receiving Organization Name), and "処理組織名" (Processing Organization Name).

101	102	105	100	103
表題	「			」
処理内容				
「				
」				
処理状況				
「				
」				
104				
受付番号	「000000」			
受付組織名	「	」	受付年月日	「000000」
処理組織名	「	」	処理年月日	「000000」

(6)

【図3】

